(2)

NEW YORK UNIVERSITY
Courant Institute of Mathematical Sciences

ALGORITHMS and TECHNIQUES for PARALLEL COMPUTERS

Contract N00014-85-K-0046

FINAL TECHNICAL REPORT

by

Principal Investigator Zvi Kedem

Period: October 15, 1984 - November 30, 1989

DEPARTMENT OF THE NAVY
OFFICE OF THE CHIEF OF NAVAL RESEARCH
800 North Quincy Street
Arlington, Virginia 22217-5000

Final Technical Report

by Zvi Kedem

Statement A per telecon
Dr. Andre Van Tilborg ONR/Code 1133
Arlington, VA 22217-5000

NWW 6/26/92

DESIGN OF PARALLEL ALGORITHMS

Many new paradigms and techniques were developed to aid the design of of parallel algorithms. The results include: the deterministic coin tossing method for breaking symmetries [5], the cascading divide and conquer paradigm which gave a new algorithm for parallel sort [1,2,3,4], the Ear-Decomposition search method [11] for graphs, the strongest results so far for parallel graph connectivity [6], finding lowest common ancestors in trees [13], two methods for parallel string matching [7,13] and methods for string matching in the presence of errors [8,9,10].

[1] M. Atallah, R. Cole, and M. Goodrich: "Cascading divide and conquer: a technique for designing parallel algorithms," SIAM Journal on Computing, 3(1989), 499-532.

[2] M. Atallah, R. Cole, and M. Goodrich: "Cascading divide and conquer: a technique for designing parallel algorithms," Twenty Eighth Annual Symposium on the Foundations of Computer Science, 1987, 151-160.

[3] R. Cole: "Parallel merge sort," SIAM Journal on Computing, 4(1988), 770-785.

[4] R. Cole and M. Goodrich: "Optimal parallel algorithms for polygon and point set problems," Fourth Annual Symposium on Computational Geometry, 1988, 201-210.

[5] R. Cole and U. Vishkin: "Deterministic coin tossing with applications to optimal parallel list ranking," Information and Control 70 (1986), 32-53.

[6] R. Cole and U. Vishkin: "Approximate parallel scheduling. II. Applications to optimal parallel graph algorithms in logarithmic time," Information and Computation 91,1 (1991), 1-47.

[7] Z. Kedem, G. Landau, and K. Palem: "Optimal Parallel Suffix-Prefix Matching Algorithm and Applications," 1st ACM Symposium on Parallel Algorithms and Architectures (Santa Fe, June 1989), pp. 388-398.

[8] G. Landau and U. Vishkin: "Efficient string matching with k mismatches," Theoretical Computer Science 43 (1986), 239-249.

[9] G. Landau and U. Vishkin: "Fast string matching with k differences," J. of Computer Systems and Sciences (special issue of selected papers from FOCS 1985) 37,1 (1988), 63-78.

[10] "G. Landau and U. Vishkin: Fast parallel and serial approximate string matching" J. of Algorithms 10 (1989), 157-169.

[11] Y. Maon, B. Schieber and U. Vishkin: "Parallel ear decomposition search (EDS) and st-numbering in graphs," Theoretical Computer Science 47 (1986), 277-298.

[12] B. Schieber and U. Vishkin: "On finding lowest common ancestors: simplification and parallelization," SIAM J. Computing 17,6 (1988), 1253-1262.

[13] U. Vishkin: "Deterministic sampling - a new technique for fast

pattern matching," SIAM J. Computing, 20,1 (1991), 22-40.

AUTOMATIC COMPILATION TO ACHIEVE FAULT-TOLERANCE IN PARALLEL COMPUTATIONS

We have created the foundation for a new field: automatic compilation of programs written for ideal models to run on realistic target architectures. The results in [1] deal with the target architectures in which processors may fail and thus are applicable to automatic generation of fault-tolerant parallel programs. Since then the results have been considerably generalized.

[1] Z. Kedem, K. Palem, and P. Spirakis: "Efficient Robust Parallel Computations," Proc. 22nd ACM Symposium on Theory of Computing, (1990), 138-148.

SUPERFAST PARALLEL ALGORITHMS

A new branch of parallel algorithmics with original computational paradigms, which deals with super fast parallel algorithms, that is those with running time of log log n or better, has been developed. Using the results developed, designers can rely on an automatic way for reducing space requirements and allocating processors. The survey paper [1] summarizes this work.

[1] U. Vishkin: "Structural parallel algorithmics," 18th Colloquium on Automata, Languages and Programming (ICALP), Barcelona, Spain, July 1991.

DISTANCE BETWEEN ORDERED TREES

Consider a distance metric based on the minimum weighted number of edits to transform one ordered tree (one in each the order of siblings matters) to another. A simple set of algorithms for this and related problems that are asymptotically more efficient than the best previous ones and efficient in practice as serial and parallel domains [1,2] These algorithms have been implemented in a toolkit running under X-Windows.

[1] D. Shasha and K. Zhang: "Simple Fast Algorithms for the Editing Distance Between Trees and Related Problems," Siam Journal on Computing, 18, 6 (1989), 1245-1262, December 1989.

[2] D. Shasha and K. Zhang: "Fast Algorithms for the Unit Cost Editing Distance Between Trees," Journal of Algorithms, 11, (1990), 581-621.

2. Significant Papers: List your 3-5 most significant publications resulting from this funding (full citations, please) and include a concise 5-10 line description of each paper and its impact (i.e., solved tough problem, opened new research area, widely cited, led to a successful commercial product, etc.)

This necessarily overlaps the previous section

R. Cole: "Parallel merge sort," SIAM Journal on Computing, 4(1988), 770-785.

This paper provides an O(log n) time O(n) processor PRAM algorithm for sorting. It is the only deterministic algorithm to achieve these bounds without requiring enormous constants in the asymptotic bounds. It was the genesis of the work on the cascading divide and conquer methodology, which was subsequently used to develop a variety of efficient parallel algorithms. The algorithm has been very widely cited.

R. Cole and U. Vishkin: "Approximate and exact parallel scheduling with applications to list, tree and graph problems," Proc. 27th Annual Symp. on the Foundations of Computer Science, 1986, 478-491.

This paper provided a framework for the design of parallel algorithms in which the needed allocation of processors changes dynamically and very quickly. In particular, it was used in the design of an asymptotically efficient and fast parallel algorithm for graph connectivity. This algorithm is the building block for many other fast and efficient parallel graph algorithms and is consequently widely cited. A practical algorithm achieving the asymptotic bounds of this paper has yet to be discovered; the role of this paper is to indicate what may be achievable.

U. Vishkin: "Structural parallel algorithmics," 18th Colloquium on Automata, Languages and Programming (ICALP), Barcelona, Spain, July 1991.

This paper surveys the development of a new branch of parallel algorithmics dealing with the design of superfast algorithms, that is those that run in time of O(log log n)$ or faster. The results have significant implications for design of parallel algorithms: (1) Designers of parallel algorithms need address space-efficiency, since an automatic way for reducing space requirements is given. (2) It is sufficient to characterize an algorithm by only specifying the operations to be performed and without allocation of these operations to processors. (3) They lead to more efficient parallel algorithms, where these super fast algorithms are used as routines.

Z. Kedem, K. Palem, and P. Spirakis: "Efficient Robust Parallel Computations," Proc. 22nd ACM Symposium on Theory of Computing, (1990), 138-148.

This paper created the foundation for a new field: automatic compilation of arbitrary programs written for ideal models to run on realistic target architectures. Thus, the issues of imperfections of the realistic machines can be shielded from applications programmers. The results of this seminal paper deal with the target architectures in which processors may fail and thus are applicable to automatic generation of fault-tolerant parallel programs. Since then the results have been considerably generalized. The field itself is growing and several research groups are currently working in it.

A. Siegel: "On universal classes of fast high performance hash functions, their time-space tradeoff, and their applications," Proc. 30th Symp. on Foundations of Computer Science, 1989, 20-25.

It has been shown that probabilistic techniques such as hashing play a major role in parallel computations. This paper shows that wildly random constant time functions are in fact programmable. Consequently, it enables many probabilistic algorithms to be proven efficient in a computable model of computation, as opposed to a model that just supposes there is some constant time mechanism that provides highly random numbers. The paper also formulates the research questions that must solved for practical implementation.

3. Presentations: Give date and forum for the 1-3 most significant presentations you gave while funded under this initiative and give a one or two sentence description of why it was important.

U. Vishkin: ""On methods for designing parallel algorithms" presented in the 1986 Workshop on "Parallel and Distributed Computation" Mathematical Sciences Research Institute, Berkeley, California.

This presentation in this workshop had a considerable impact on the research directions in the parallel non-numeric algorithms designers community.

D. Shasha "Matching Algorithms for Ordered Labelled Trees," National Cancer Institute March, 1988 Frederick, Maryland.

The significance was that this led to cooperation between our approximate tree matching group and NIH, resulting first in an implementation of approximate tree matching that they have used for approximate recognition of 2 dimensional RNA patterns. This also encouraged us to develop our approximate tree toolkit called ATBE (approximate tree by example).

==

C. PRODUCTS

Finally, we need to know of any outside recognition or tangible outcomes of the research supported with this funding. Please be as complete as possible here.

1. Honors and awards: Give date, recipient and a brief description for each honor or award received by a PI and specify which awards resulted ostensibly from this funding.

R. Cole, Guggenheim Fellowship, 1988-1989. This is a very prestigious fellowship awarded to fewer than 10 Computer Scientists per year.

This was awarded to enable the PI to pursue work on asynchronous parallel algorithms. While the award did not result from the grant directly, it is clear that it recognized the PI's work in parallel algorithms, work made possible in part by the contract

2. Software and hardware prototypes: Give a brief description of any software or hardware prototypes developed under this initiative and the significance of each. If the prototype has been distributed, tell where and how it is being used in #4 below.

The ATBE (Approximate Tree By Example) software prototype to perform approximate comparison of trees was inspired by the research although not directly funded by the contract. It currently provides the following functionality:

* Given two ordered trees, it will discover the minimum number of inserts, deletes or relabellings necessary to transform one tree to the other.

* It will solve the above problem after optimal removal of subtrees from one of the two trees.

* It will solve the first problem even in the presence of variable length don't cares.

3. Patents: List all patent applications filed (including software copyrights) resulting from this funding, give the patent application number for each one, and indicate which have been granted.

4. Major transitions: Describe any of the following that resulted from this funding:

b. Significant use of your prototypes, algorithms, ideas, etc. in other research projects.

1: Parallel Algorithmics

The research on a core of parallel algorithmic design paradigms conducted as the main focus of research by U. Vishkin and as a major focus of research by other participants had has enormous impact on the field. To keep this part short, we will refer only to selected review papers and a textbook. These review papers describe numerous transitions of these works into the research of other researchers.

* A survey paper by Richard M. Karp and Vijaya L. Ramachandran: "Parallel algorithms for shared-memory machines," in Handbook of Theoretical Computer Science - Volume A: Algorithms and Complexity, editor J. van Leeuwen, MIT Press, 1990, pp. 871-941, references numerous research results funded by the contract. This handbook "was designed to provide a wide audience of professionals and students in Computer Science and related disciplines with an overview of the major results and developments in the theoretical exploration of issues in computing and information processing to date."

* A survey paper by David Eppstein and Zvi Galil: "Parallel algorithmic techniques for combinatorial computations" in Ann. Rev. Comput. Sci. 1988,3: 233-283.

* The 1992 textbook by Joseph JaJa "An Introduction to Parallel Algorithms," Addison-Wesley.

* A 1991 report by F. Abolhassan, J. Keller and W.J Paul (Universitat des Saarabrucken, West Germany) on the cost-effectiveness and realization of the theoretical PRAM model, is another concrete example on the transition of research ideas due to U. Vishkin. This last report extends the Fluent machine design by Ranade (UC-Berkeley) and provides additional evidence (on paper only at this stage) that a "virtual PRAM" is an excellent candidate for detailed engineering prototyping. Since then major funding was awarded to W.J. Paul for building such a machine.

2: Automatic Transformation of Ideal Programs for Execution on Realistic Architectures.

Following the seminal paper by Z. Kedem, K. Palem, and P. Spirakis: "Efficient Robust Parallel Computations," Proc. 22nd ACM Symposium on Theory of Computing, (1990), 138-148, worked continued on this topic. Current results due to Z. Kedem, K. Palem, M. Rabin, and A. Raghunathan present a completely general method for automatically transforming programs written for an ideal model of a perfect synchronous machine into programs that can execute on a faulty asynchronous machine. The transformations has the following essential properties:

* Reliability: The method was provably correct in that the computational semantics of the given ideal (application) program were preserved under the transformation. That is, the execution was reliable.

* The method generated reliable programs with small loss of efficiency compared to the original ideal programs. Furthermore, the loss of efficiency scales with the size of the computation and the reliability of the target machine.

* Potential for automatic compilation: The method is fully automatic (not requiring any semantic knowledge of the application) and

therefore suitable for implementation as a module in a compiler thus freeing the application programmer from concern with the target's machine imperfections.

The seminal paper and the follow-up research has generated considerable interest in the research community. Currently two additional research groups, one at the University of California and the other at Hebrew University are examining various additional issues in this field.

3: Approximate Tree Matching

The ATBE software prototype (motivated on research of D. Shasha) is used by various research groups in fields ranging from molecular biology to sociology. All these uses are preliminary at this stage.